

## Financial Time Series Forecasting with Grouped Predictors using Hierarchical Clustering and Support Vector Regression

ZheGao<sup>a,b,\*</sup> and JianjunYang<sup>c</sup>

a. *Guanghua School of Management, Peking University, Beijing 100871, China*

b. *Peking University Founder Group, Beijing, 100871, China*

c. *Department of Intelligence Science, Center for Information Science, Room 2314, Science Building 2, Peking University, Beijing 100871, China*  
truman.gz@gmail.com (ZheGao), yjjhyxy@gmail.com (Jianjun Yang)

### Abstract

*Financial time series prediction is regarded as one of the most challenging tasks due to the inherent noise and non-stationality of the data. This paper proposed a two-stage financial time series prediction approach hybridizing support vector regression (SVR) with hierarchical clustering (HC). By averaging the variables within the clusters obtained from hierarchical clustering, we define super predictors and use them as the input variables of the SVR forecasting model. Although averaging is a simple technique, it plays an important role in reducing variance. To evaluate the performance of the proposed approach, the Shanghai-Shenzhen 300 index is used as illustrative example. The experimental results show that the proposed approach outperforms both the SVR model with principal component analysis and the SVR model with genetic algorithms in average prediction error and prediction accuracy.*

**Keywords:** *Hierarchical clustering, Support vector regression, Financial time series forecasting*

### 1. Introduction

The financial market is one of the most complicated systems due to its undergoing constant evolution and non-linear dynamics. Financial time series forecasting is regarded as one of the most challenging tasks. Over the last two decades, this subject has attracted numerous attention from investors, speculators and researchers [28, 29], and it has become an important issue in investment decision making.

Recently, growing attention has been paid to the use of large datasets for financial time series forecasting. Since both the availability of financial time series data and the computational power to handle them have increased tremendously, researchers are trying to enrich their forecasting models by taking advantage of a broader information base. Conventional time series models are not suitable for incorporating a large number of predictors, so new methods were developed. Since the existence of significant residual variance of the actual stock return from the prediction of the regression equation, it is possible that nonlinear models are suitable to explain this residual variance. Recently, the support vector machine (SVM), based on statistical learning theory [13], has led to great potential and superior performance in a range of applications, including financial stock market prediction [2, 7, 12, 4, 23-26]. SVM is shown to be very resistant to the over-fitting problem and can

---

\*Corresponding author. Tel.: (86)-10-68588329; (86)13520131418

achieve a high generalization performance comparison to artificial neural networks. This is largely due to the unique theory of the structural risk minimization principle used to estimate a function by minimizing an upper bound of the generalization error. The solution of SVM, whose training process is equivalent to solving a constrained quadratic programming problem, is always unique and globally optimal. The weak points of neural networks, such as over-fitting and getting stuck at local minima, have been solved by SVM in theory. Support vector regression (SVR), the regression model of SVM, deals with continuous data and has been receiving increasing attention to solve nonlinear regression problems [1, 7, 19, 20-22, 27].

In the prediction of financial time series, another key problem is the inherent noise of the predictors, especially when the number of predictors is large. Learning SVR from the raw data with noise may torture the approximation function and result in the loss of generalization capability in the testing step. This is because not all of the pre-chosen predictors are informative or can provide high discrimination power. This is known as the curse of dimensionality problem [11]. In general, some related work considers a feature selection step to examine the validity of their chosen variables for effective stock prediction. Feature selection can lead to many potential benefits, such as reducing data complexity and training time, etc. There are several feature selection methods, such as filter-based feature selection (FBFS) [1], genetic algorithms (GA) [14], principal component analysis (PCA) [6] and independent component analysis (ICA) [7]. However, in most applications, these methods have been applied to relatively small systems and that the relationship structure among the large time series have not been given enough attention in the literature.

In this paper, we present a way to improve the generalization capability of SVR through coupling with the hierarchical clustering method, and applied it to financial time series forecasting. We first use hierarchical clustering on the predictors to obtain a dendrogram that reveals their nested correlation structure. At each level of the hierarchy, we create a unique set of predictors and super predictors by computing the average expression of the current clusters. The treatment here can overcome the collinearity problem among the predictors, which results in large variance of the estimates and inaccurate prediction [9]. Then we used the different sets of predictors and super predictors as inputs for SVR. The empirical results show the efficiency of the proposed approach.

The rest of this paper is organized as follows: Section 2 gives a brief introduction about hierarchical clustering and support vector regression. The proposed hybrid forecasting model is thoroughly described in Section 3. The experimental setup, including the chosen dataset and the process of constructing the prediction model and the evaluation methods is presented in Section 4. Section 5 shows the experimental results and a conclusion is provided in Section 6.

## 2. Methodology

### 2.1. Hierarchical Clustering

Clustering analysis, one of the primary data analysis tasks in knowledge discovery process, is used to identify valid, potentially useful, and understandable patterns in large amount of data. Hierarchical clustering is used in our approach because it provides multiple levels at which the super predictors can be formed. Hierarchical clustering has several advantages over traditional flat clustering, such as k-means. It is more robust with respect to the input parameters, less influenced by cluster shapes, less sensitive to largely differing point densities of clusters, and more informative for data display.

An agglomerative algorithm for hierarchical clustering is used in our paper. In order to decide which clusters should be combined for agglomerative, a measure of distance (or

similarity) between sets of predictors is required. In most methods of hierarchical clustering, this is achieved by the using of an appropriate metric (a measure of distance or similarity between pairs of predictors), and a linkage criterion which specifies pairwise distances of predictors in the sets. Correlation distance is used in our paper and is defined on couples of time series (x,y) as:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{2(1 - c_{xy})} \quad (1)$$

Where  $\mathbf{x} = (x_1, x_2, \dots, x_T)$ ,  $\mathbf{y} = (y_1, y_2, \dots, y_T)$  and  $c_{xy}$  is the correlation coefficient between  $\mathbf{x}$  and  $\mathbf{y}$ .

There are several commonly used linkage criteria, such as single linkage, complete-linkage, average-linkage and minimum energy clustering. We use the average-linkage clustering and consider the distance between one cluster and another to be equal to the average distance. Given a set of N predictors and an N×N similarity matrix, the basic process of hierarchical clustering is implemented as follows [8]:

1. Start by assigning each predictor to a cluster. Set the distances (similarities) between the clusters equal to the distances (similarities) between the predictors in the clusters.
2. Find the most similar pair of clusters and merge them into a single cluster.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat Steps 2 and 3 until all predictors are clustered into a single cluster of size N.

## 2.2. $\varepsilon$ -Support Vector Regression

For illustrating the concept of SVR, soft margin support vector regression ( $\varepsilon$ -SVR) is explained and the nonlinear case is discussed in this section. Consider a set of training data  $D = \{x_i, y_i\}_{i=1}^n$ , in which  $x_i \in R^d$  is the input vector,  $y_i$  represents the corresponding scalar output, and n is total number of the data patterns. The objective of linear SVR is to build the regression function formulated as:

$$f(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b; \mathbf{w} \in R^d, b \in R \quad (2)$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product in the input space. A loss function, (Eq. 3) known as a  $\varepsilon$ -insensitive function is usually used to evaluate the robustness of SVR.

$$L_\varepsilon(f(x_i), y_i) = \begin{cases} 0 & \text{if } |f(x_i) - y_i| \leq \varepsilon \\ |f(x_i), y_i| - \varepsilon & \text{otherwise} \end{cases} \quad (3)$$

The coefficients  $\mathbf{w}$  and  $b$  are estimated by minimizing

$$R = \frac{1}{2} \|\mathbf{w}\|^2 + C \frac{1}{n} \sum_{i=1}^n L_\varepsilon(f(x_i), y_i) \quad (4)$$

In the risk function given by Eq. (4), the first term is the regularization term and the second term  $C(1/n) \sum_{i=1}^n L_\varepsilon(f(x_i), y_i)$  is the empirical risk, which is measured by the  $\varepsilon$ -insensitive loss function given by Eq. (3).  $C$  is referred to as the regularized constant and is used to specify the trade-off between the regularization term and the empirical risk. Both  $C$  and  $\varepsilon$  are pre-defined parameters.

The deviation  $L_\varepsilon(f(x_i), y_i)$  from the boundaries of the  $\varepsilon$ -insensitive zone can be measured by two positive slack variables,  $\xi_i$  and  $\xi_i^*$ ,  $i = 1, 2, \dots, n$ . By introducing the slack variables, Eq. (4) is transformed into the following constrained form:

$$\begin{aligned}
 \text{Minimize } R(\mathbf{w}, \xi, \xi^*) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) & (5) \\
 \text{Subjected to } y_i - f(x_i) &\leq \varepsilon + \xi_i, \\
 f(x_i) - y_i &\leq \varepsilon + \xi_i^*, \\
 \xi_i &\geq 0, \xi_i^* \geq 0.
 \end{aligned}$$

By applying the Lagrangian theory and the Karush Kuhn-Tucker (KKT) condition to this constrained optimization problem, it yields the corresponding dual form as follows:

$$\begin{aligned}
 \text{Maximize } -\varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle & (6) \\
 \text{Subjected to } \mathbf{e}^T ((\alpha_i - \alpha_i^*)) &= 0, \\
 0 \leq \alpha_i, \alpha_i^* \leq C, i &= 1, 2, \dots, n
 \end{aligned}$$

where  $\mathbf{e} = [1, \dots, 1]^T$  is the vector of all ones, and  $\alpha_i, \alpha_i^*$  are Lagrange parameters. Eq. (6) represents a standard constrained quadratic programming problem, for which the common solution method is sequential minimal optimization (SMO) algorithm [10]. By solving problem (6), the approximate function becomes:

$$\hat{f}(x) = \sum_{i,j=1}^n (-\alpha_i + \alpha_i^*) \langle x_i, x \rangle + b \quad (7)$$

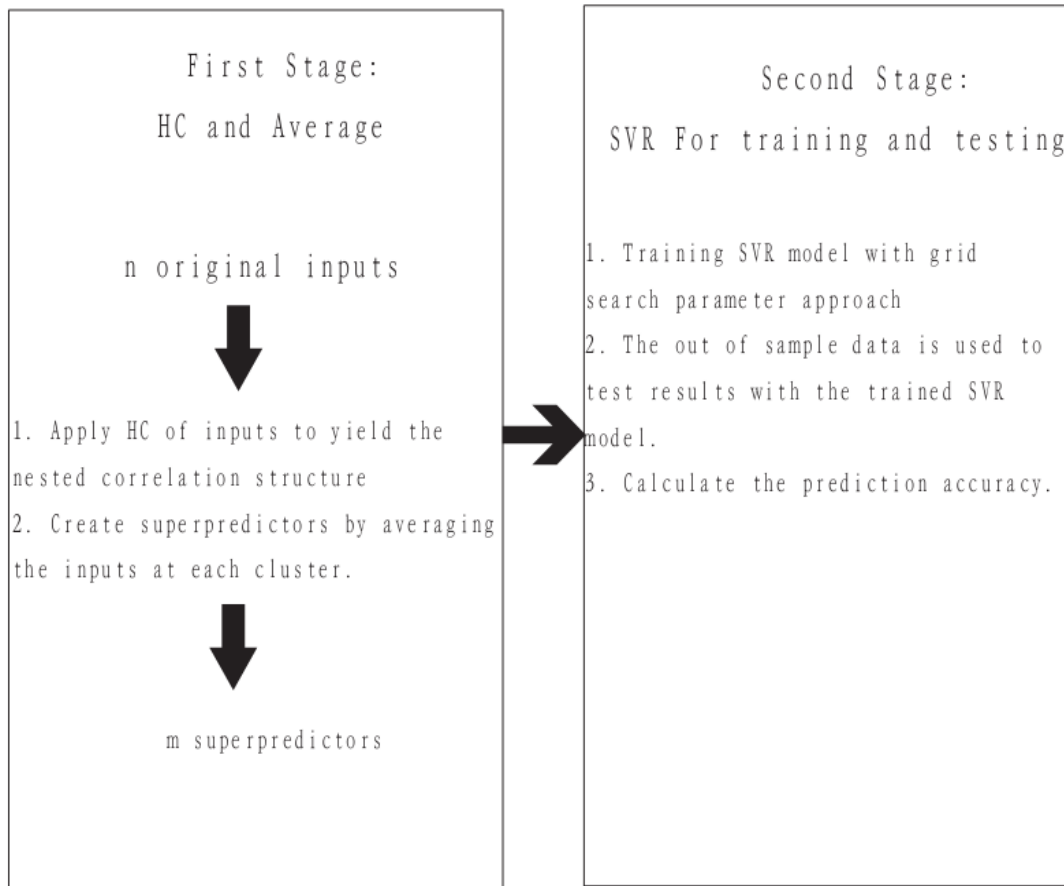
The next step is to extend the SVR algorithm to the nonlinear problem. This can be done by mapping the training patterns  $x_i$  into a higher-dimensional feature space via a map function  $\mathcal{O}$ . We could use an implicit mapping via kernels, which is computationally cheaper. As noted above, the SVR algorithm only depends on dot products between patterns  $x_i$ , which can be replaced by scalar products  $\langle \mathcal{O}(x_i), \mathcal{O}(x_j) \rangle$ . The scalar product is calculated directly by computing a kernel function  $K(x_i, x_j)$  to avoid performing a mapping  $\mathcal{O}(x)$ . The radial basis function (RBF) is used in our paper, which is defined as  $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ , where  $\gamma$  denotes the width of the RBF.

### 3. The Proposed System Framework

In this study, a two-stage hybrid method is implemented which combines the SVR prediction capability with the hierarchical clustering algorithm. In the first stage, HC clusters the original inputs to yield the nested correlation structure. At each level of hierarchy, the correlated variables are automatically collected and the super predictors are created by averaging the variables at each cluster. This gives  $p$  different sets of super predictors that represent each level. In the second stage, an individual SVR model for each set of the super predictors is constructed. In order to construct an accurate SVR model, the grid algorithm is adapted to find the proper parameters. Figure 1 shows detailed steps to train and test the two stage HC-SVR model.

Our method is advantageous when there are multiple variables with strong correlations. Specifically, if the sample correlations of the predictors are high enough, an averaged predictor lead to lower expected squared error of coefficient estimates. This fact has been proven by [9], who used the averaged features as inputs for Lasso regression. By averaging

the original variables, the SVR model in our method yields a more accurate result than the individual predictors.



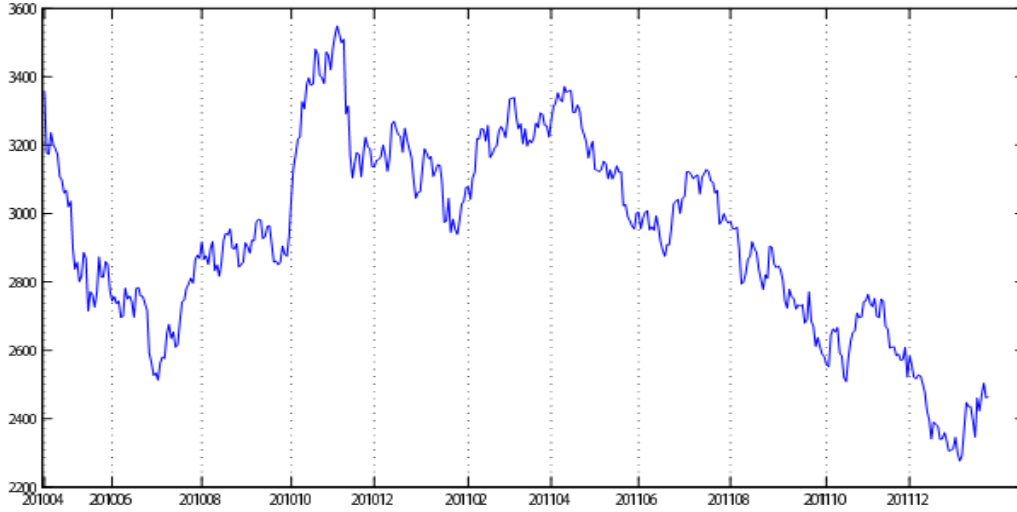
**Figure 1. The Hybrid HC-SVR Architecture**

## 4. Experiment Design

### 4.1. Datasets and Variables

For evaluating the performance of the proposed hybrid HC-SVR model, the Shanghai-Shenzhen 300 index (CSI300) of China is used in this study. The CSI300 selects 300 sweeping, liquid and most representative constituent stocks listed on the Shanghai and Shenzhen Stock Exchanges and is the first joint release of the A share market index.

Daily data from April 16, 2010 to January 31, 2012 with 433 observations are used in the evaluation, as shown in Figure 2. The sliding window is used to divide the sample data into different groups of training and testing data. The training set is composed of 10 months of samples and the testing set is based on the following month. For example, the training data of the first group is from the 20100416 to 20101031 and the testing data is from 20101101 to 20101130. Therefore, the model is trained and tested for 12 times.



**Figure 2. The CSI300 from January 5 to January 31, 2012**

For forecasting the CSI300 closing prices, two groups of totally 26 indicators are used as forecasting variables. The first group are the 4 CSI300 indexfutures contracts traded on China Financial Futures Exchange (CFFE). Since the futures prices usually change before the market price [5], using the leading futures as inputs should contribute to the success in increasing the forecasting accuracy. The second group are 22 technical indicators determined by the review of domain experts and literatures, which are the most widely used features in financial time series prediction. The descriptions and definitions of these input variables are presented in Appendix A.

**4.2. Evaluation Strategies**

The prediction performance is evaluated by four performance measures, namely, the root mean square error (RMSE), normalized mean square error (NMSE), mean absolute error (MAE), and directional symmetry (DS) (Table 1). RMSE, NMSE and MAE are measures of the deviation between actual and predicted values. The smaller values of RMSE, NMSE, MAE, the closer are the predicted time series values to that of the actual value. DS provides the correctness of the predicted direction of the CSI300 index in terms of percentage. A large value of DS suggests a better predictor.

**Table 1. The Definitions of the Performance Measures**

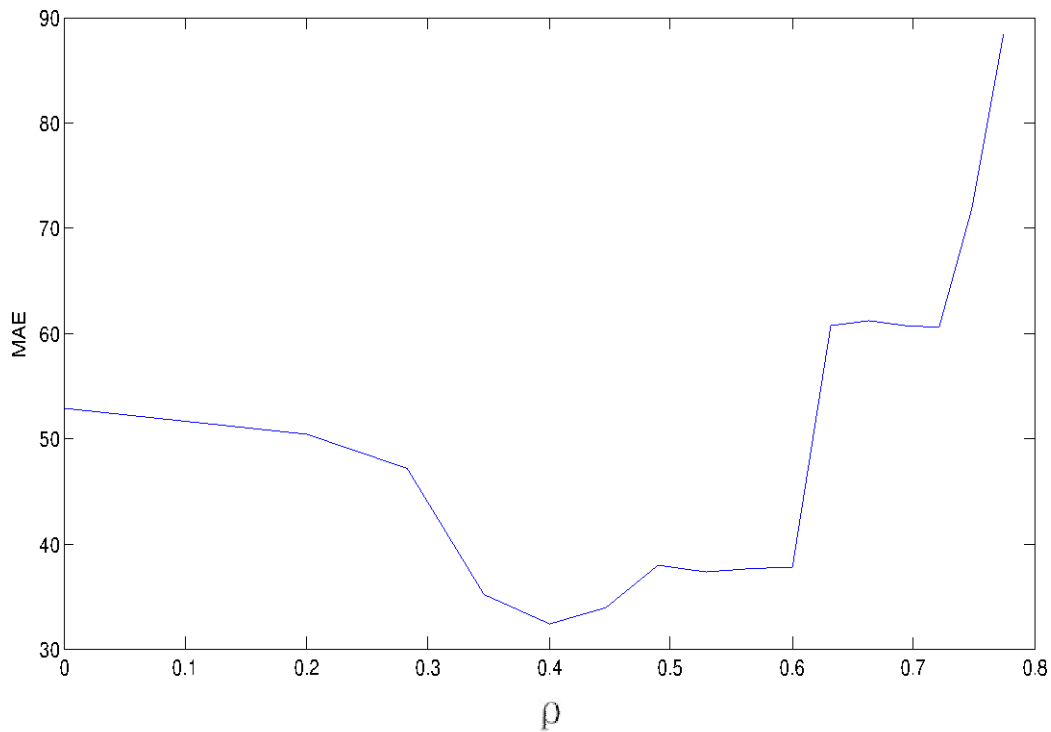
Metrics	Definition
RMSE	$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$
NMSE	$NMSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sigma^2 n}, \text{ where } \sigma^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}$
MAE	$MAE = \frac{\sum_{i=1}^n  y_i - \hat{y}_i }{n}$

DS	$DS = \frac{100}{n} \sum_{i=1}^n d_i, \text{ where } d_i = \begin{cases} 1 & (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$
----	---

## 5. Experiment Results

### 5.1. Determining the Level of HC

Each level of hierarchical tree shows different performance. To determine the best level of HC, we use a threshold  $\rho$ , which is the minimum distance among the clusters in that level. The parameter  $\rho$  affects the granularity of clustering and then indirectly impacts on the goodness of the fit. To investigate the impacts of the number of clusters on the MAE performance, we construct the models for different value  $\rho$ . Figure 3 shows the MAE of HC-SVR at various  $\rho$ , in which  $C$  and  $\epsilon$  are respectively, fixed at 20 and 0.0005. The figure shows that the MAE on the testing set decreases when  $\rho$  increases from 0 to 0.4, but subsequently increases while  $\rho$  increases. This indicates that too large a value of  $\rho$  (0.5-0.8) yields a larger bias to the solution of SVR. In the following experiments, we set  $\rho$  to 0.4.



**Figure 3. The Behavior of MAE in HC-SVR the MAE on the Testing Set Decreases when  $\rho$  Increases from 0 to 0.4, but Subsequently Increases while  $\rho$  Increases**

### 5.2. Comparison

This study compares the prediction accuracy of the HC-SVR method with PCA-SVR and GA-SVR, as PCA and GA are popular methods to select the features. The differences among these three algorithms are firstly compared, which produces the inputs to the SVR model. Then we use the same RBF kernel,  $\epsilon$ -insensitive loss function and the grid search approach.

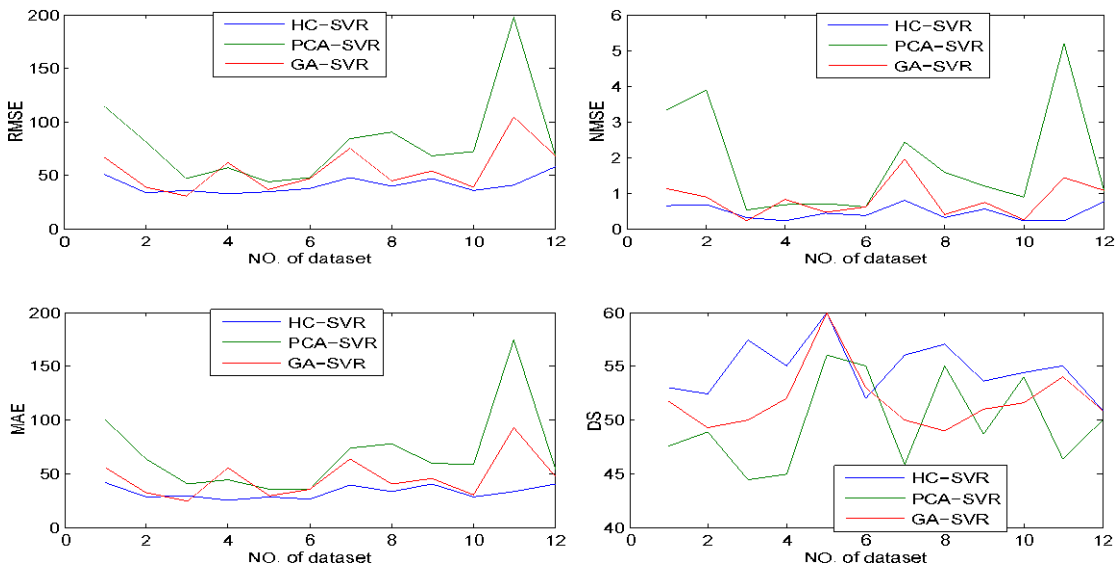
To perform PCA-SVR, the question then is, how many factors do we need to extract various guidelines have been developed, and in this paper we use the Kaiser criterion [3]. Specifically, we retain only factors with eigenvalues greater than 1, which means that unless a factor extracts at least as much as the equivalent of one original variable, we drop it. In our example, using this criterion we would retain 6 principal components.

To perform GA-SVR, we use the grid search algorithm to select the best possible results in terms of MAE. According to the prediction performance, the parameters used in this paper for comparisons are as follows: the population size is set to 25, the crossover rate is set to 0.5 and mutation rate is set to 0.03.

The performances of HC-SVR, PCA-SVR and GA-SVR on the test set are collated and the average results of 12 test sets are summarized in Table 2. It can be seen that HC-SVR has smaller RMSE, NMSE and MAE but larger DS than PCA-SVR, GA-SVR. This indicates that HC-SVR outperforms PCA-SVR and GA-SVR in predicting CSI300, and it can forecast more accurately and capture the turning points better than the others. To evaluate the robustness of the proposed HC-SVR method, Fig. 4 shows the RMSE, NMSE, MAE and DS for the three models at each test set. As we can see, for most of the test sets, the proposed HC-SVR method outperforms the other benchmarking tools. It therefore indicates that HC-SVR based approach indeed provides better forecasting accuracy than the other two approaches.

**Table 2. Averaged Results of HC-SVR, PCA-SVR and GA-SVR on the Test Set**

	RMSE	NMSE	MAE	DS
HC-SVR	41.079	0.018	32.686	55.319
PCA-SVR	91.388	0.091	68.656	48.319
GA-SVR	59.309	0.038	46.485	51.479



**Figure 4. RMSE, NMSE, MAE, DS Comparisons: HC-SVR vs. PCA-SVR vs. GA-SVR\***

\*Note: The performances of HC-SVR, PCA-SVR and GA-SVR on the test set are collated and the average results



## 6. Conclusions

In this study, we proposed a two-stage forecasting model by integrating HC and SVR for financial time series prediction. We first use hierarchical clustering to obtain the sets of correlated variables; then we average the variables within each cluster and input the averages as superpredictors to SVR. When the variables are highly correlated, their average becomes a strong feature, yielding a fit with lower variance than the individual variables. To forecast CSI300, this study compares the proposed method with PA-SVR and GA-SVR using prediction error and prediction accuracy as the criteria. Experimental results show that the proposed model can produce lower prediction error and higher prediction accuracy and outperformed the PA-SVR and GA-SVR.

Many problems in finance require the exploitation of large panels of time series. The idea proposed in this study can be extended to handle the large-dimensionality problem which could be solved with the development of parallel computing [15-18]. Future researches may integrate HC-SVR with other signal processing techniques, like nonnegative matrix factorization to improve the forecasting capabilities.

## Appendix A. Technical Indicator List

1. **OP,HP,LP,CP,VOL** the intraday opening price, high price, low price, close price and volume, respectively
2. **MA(CP,n)** n-day moving average, calculated as  $MA(CP,n)_t = \sum_{i=t-n+1}^t CP_i / n$
3. **EMA(CP,n)** n-day exponential moving average, calculated as  $EMA(CP,n)_t = (4 * EMA(CP,n)_{t-1} + CP_t) / n$
4. **RSI(CP,n)** n-day relative strength index, calculated as  $RSI(CP,n)_t = 100 \times EMA(U,n)_t / (EMA(U,n)_t + EMA(D,n)_t)$ , where U, D are respectively upward and downward change for each trading period.
5. **CCI(n)** n-day commodity channel index, calculated as  $CCI(n) = (p_t - MA(p,n)_t) / (0.015 * \sigma(p_t))$ , where the  $p_t$  is the typical price, and  $\sigma$  is the mean absolute deviation.
6. **PSY(CP,n)** n-day psychological line, calculated as  $PSY(CP,n)_t = 100 \times UD_n / n$ , where  $UD_n$  is the number of upward days during previous n days.
7. **VR(n)** n-day volume ratio, calculated as  $VR(n)_t = 100 \times (TH * 2 + TQ) / (TL * 2 + TQ)$ , where  $TH, TL, TQ$  are respectively the volume summation at upward days, downward days, and non-trend days.
8. **WR(n)** n-day Larry Williams' oscillator, calculated as  $WR(n)_t = (HHV(HP,n) - CP_t) / (HHV(HP,n) - LLV(LP,n))$ , where

---

of 12 test sets are summarized in Table 2. It can be seen that HC-SVR has smaller RMSE, NMSE and MAE but larger DS than PCA-SVR, GA-SVR. This indicates that HC-SVR outperforms PCA-SVR and GA-SVR in predicting CSI300, and it can forecast more accurately and capture the turning points better than the others. To evaluate the robustness of the proposed HC-SVR method, Fig. 4 shows the RMSE, NMSE, MAE and DS for the three models at each test set. As we can see, for most of the test sets, the proposed HC-SVR method outperforms the other benchmarking tools. It therefore indicates that HC-SVR based approach indeed provides better forecasting accuracy than the other two approaches.

$HHV, LLV$  are respectively the highest high price and lowest low price in previous  $n$  days period.

9. **BIAS(n)**  $n$ -day BIAS, calculated as

$$BIAS(n)_t = 100 \times (CP_t - MA(CP, n)) / MA(CP, n)$$

10. **AR(n)**  $n$ -day A ratio, calculated as

$$AR(n)_t = \sum_{i=t-n+1}^t (HP_i - O_i) / \sum_{i=t-n+1}^t (OP_i - LP_i)$$

11. **BR(n)**  $n$ -day B ratio, calculated as

$$BR(n)_t = \sum_{i=t-n+1}^t (HP_i - CP_{i-1}) / \sum_{i=t-n+1}^t (CP_{i-1} - LP_i)$$

12. **K(n)**  $n$ -day stochastic index K, calculated as

$$K(n)_t = (n-1) / n \times K(n_{t-1}) + 1 / n \times RSV(n)_t, \text{ where}$$

$$RSV(n)_t = 100 \times (CP_t - LLV(LP, n)) / (HHV(HP, n) - LLV(LP, n))$$

13. **D(n)**  $n$ -day stochastic index D, calculated as

$$D(n)_t = (n-1) / n \times D(n)_{t-1} + 1 / n \times K(n)_t$$

14. **DIF(short, long)** the difference between a short-day and long-day exponential moving average, calculated as

$$DIF(short, long)_t = EMA(CP, short)_t - EMA(CP, long)_t$$

15. **DEA(n)**  $n$ -day exponential moving average of  $DIF$ , calculated as

$$DEA(n)_t = EMA(DIF, n)$$

16. **MACD** moving average convergence/divergence, calculated as

$$MACD_t = 2 \times (DIF - DEA)$$

17. **ROC(n)**  $n$ -day price change rate of change, calculated as

$$ROC(n)_t = 100 \times (CP_t - CP_{t-n}) / CP_{t-n}$$

18. **MTM(n)**  $n$ -day momentum, calculated as  $MTM(n)_t = CP_t - CP_{t-n}$

## References

- [1] C. Huang and C. Tsai, "A hybrid softm-SVR with a filter-based feature selection for stock market forecasting", *Expert Systems with Applications*, vol. 36, no. 2, (2009), pp. 1529–1539.
- [2] W. Huang, Y. Nakamori and S. Wang, "Forecasting stock market movement direction with support vector machine", *Computers & Operations Research*, vol. 32, no. 10, (2005), pp. 2513–2522.
- [3] H. Kaiser, "The varimax criterion for analytic rotation in factor analysis", *Psychometrika*, vol. 23, no. 3, (1958), pp. 187–200.
- [4] K. Kim, "Financial time series forecasting using support vector machines", *Neuro computing*, vol. 55, no. 1-2, (2003), pp. 307–319.
- [5] T. Lee and N. Chen, "Investigating the information content of noncash-trading index futures using neural networks", *Expert Systems with Applications*, vol. 22, no. 3, (2002), pp. 225–234.
- [6] X. Lin, Z. Yang and Y. Song, "Short-term stock price prediction based on echo state networks", *Expert Systems with Applications*, vol. 36, no. 3, (2009), pp. 7313–7317.
- [7] C. Lu, T. Lee and C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression", *Decision Support Systems*, vol. 47, no. 2, (2009), pp. 115–125.
- [8] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms", *The Computer Journal*, vol. 26, no. 4, (1983), pp. 354–359.
- [9] M. Park, T. Hastie and R. Tibshirani, "Averaged gene expressions for regression", *Biostatistics*, vol. 8, no. 2, (2007), pp. 212–227.
- [10] J. Platt, *et al.*, "Sequential minimal optimization: A fast algorithm for training support vector machines", (1998).

- [11] M. Kaufmann, C. Saunders, A. Gammerman and V. Vovk, "Ridge regression learning algorithm in dual variables", In: Proceedings of the 15th International Conference on Machine Learning, vol. 19980, (1998), pp. 515–521, San Francisco, CA, USA.
- [12] F. Tay and L. Cao, "Application of support vector machines in financial time series forecasting", OMEGA-OXFORD-PERGAMON PRESS, vol. 29, (2001), pp. 309–317.
- [13] V. Vapnik, "An overview of statistical learning theory", Neural Networks, IEEE Transactions on, vol. 10, no. 5, pp. 988–999.
- [14] L. Yu, S. Wang and K. Lai, "Mining stock market tendency using GA-based support vector machines", Internet and Network Economics, (2005), pp. 336–345.
- [15] Z. Jilin, L. Enyi, W. Jian, R. Yongjian, Y. Miao and W. Jue, "Implementing Sparse Matrix-Vector Multiplication with QCSR on GPU", APPLIED MATHEMATICS & INFORMATION SCIENCES VOL, vol. 7, no. 2, (2003), pp. 473-482.
- [16] J.-L. Zhang, W. Qian, X.-H. Xu, J. Wan, Y.-Y. Yin and Y. J. Ren, "WLBS: A weight-based metadata server cluster load balancing strategy", International Journal of Advancements in Computing Technology, vol. 4, no. 1, (2004), pp. 77-85.
- [17] J. L. Zhang, J. H. Zhang, H. Ding, J. Wan, Y. J. Ren and J. Wang, "Designing and Applying an Education IaaS System based on OpenStack", APPLIED MATHEMATICS & INFORMATION SCIENCES, vol. 7, no. 2, (2007), pp. 155-160.
- [18] J.-L. Zhang, Y. Chen, J. Wang, E.-Y. Liu, Y.-Y. Yin, Y.-J. Ren and C.-H. Li, "The implementation and optimization of irregular application task models based on the cell BE processor", International Journal of Digital Content Technology and its Applications, vol. 6, no. 2, (2012), pp. 269-278.
- [19] C.-J. Lu and Y.-W. Wang, "Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting", International Journal of Production Economics, vol. 128, Issue 2, (2010) December, pp. 603-613.
- [20] B. Wang, H. Huang and X. Wang, "A novel text mining approach to financial time series forecasting", Neuro computing, vol. 83, (2012) April 15, pp. 136-145.
- [21] C.-S. Lin, S.-H. Chiu and T.-Y. Lin, "Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting", Economic Modeling, vol. 29, Issue 6, (2012) November, pp. 2583-2590.
- [22] Y. Wang, B. Wang and X. Zhang, "A New Application of the Support Vector Regression on the Construction of Financial Conditions Index to CPI Prediction", Procedia Computer Science, vol. 9, (2012), pp. 1263-1272.
- [23] K.-J. Kim, "Financial time series forecasting using support vector machines", Neuro computing, ISSN 0925-2312, vol. 55, Issues 1–2, (2003) September, pp. 307-319.
- [24] E. Francis, H. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting", Neuro computing, vol. 48, Issues 1–4, (2002) October, pp. 847-861.
- [25] C.-J. Lu, T.-S. Lee and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression", Decision Support Systems, vol. 47, Issue 2, (2009) May, pp. 115-125.
- [26] S. Ismail, A. Shabri and R. Samsudin, "A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting", Expert Systems with Applications, vol. 38, Issue 8, (2011), August, pp. 10574-10578.
- [27] Y. Ding, X. Song and Y. Zen, "Forecasting financial condition of Chinese listed companies based on support vector machine", Expert Systems with Applications, vol. 34, Issue 4, (2008), May, pp. 3081-3089.
- [28] Z. Gao, J. Yang and S. Tan, "Forecasting Chinese GDP with mixed frequency data set: A generalized lasso granger method", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS, vol. 7929, no. 2, (2013), pp. 163-172.
- [29] Z. Gao, Z. Wang, L. Wang and S. Tan, "Linear non-Gaussian causal discovery from a composite set of major US macroeconomic factors", Expert Systems with Applications, vol. 39, Issue 12, (2012) September 15, pp. 10867-10872.

